Manual Técnico

Organización de Lenguajes y Compiladores 1

José Rafael Morente González | 201801237
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Contenido

Análisis Léxico

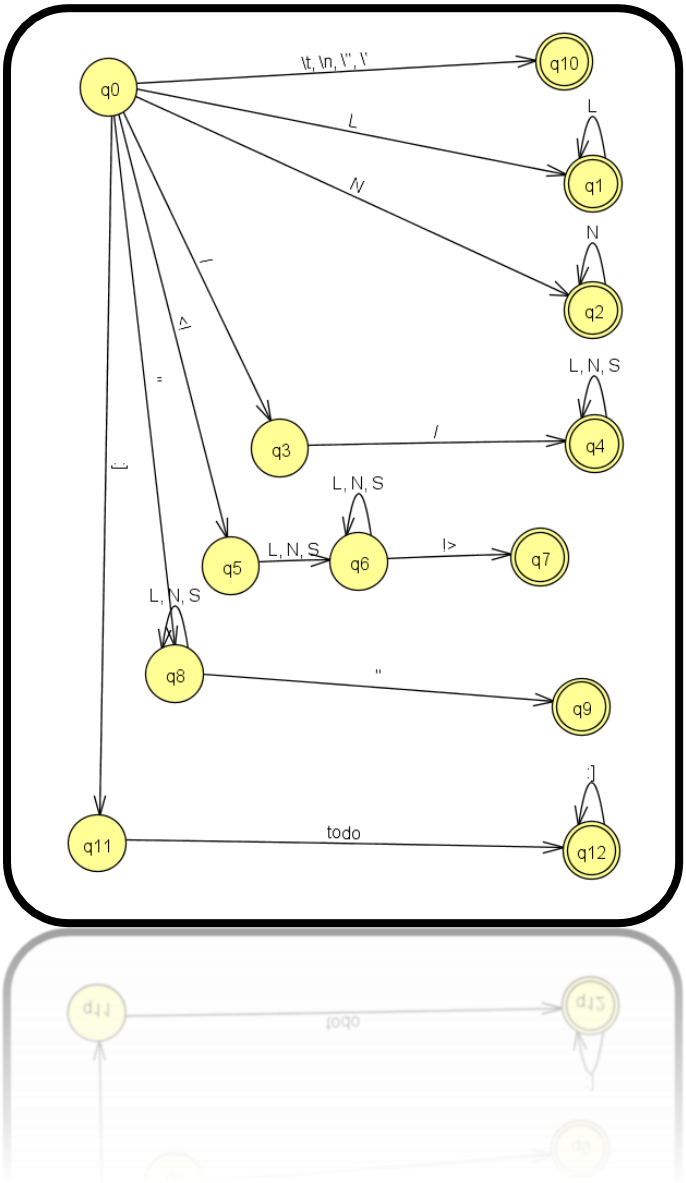
Conjuntos

$L = \{a, A, b, B, \dots, \dots, z, Z\}$

$S = \{Todos\ los\ Signos\}$

$\underline{N = \{0,1 \dots ,9\}}$

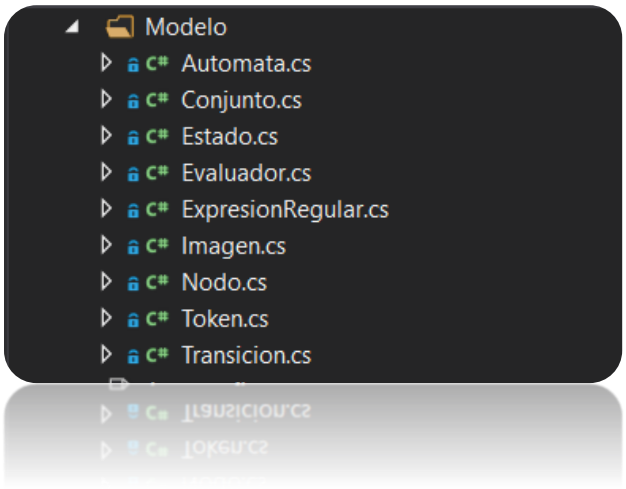
AFD



Método de Thompson

A continuación, se presenta una breve explicación de la generación del método de Thompson.

Modelos



- Autómata: Es el modelo que representa el autómata con sus estados de inicio fin aceptación y estados con los que tiene transiciones.
- Conjunto: Es el modelo que nos permite guardar los elementos del conjunto que se presentan al inicio.
- Estado: Representa un estado en el autómata y contiene una lista de transiciones.
- Transición: Tiene un inicio un fin y el símbolo con el que se realiza dicha transición.
- Nodo: Este modelo nos ayudara a construir un árbol binario el cual nos ayudara a ordenar de una manera eficaz la expresión regular.

Método de Thompson

El método de Thompson está basado en la creación de Autómatas Finitos No Deterministas. El mismo se genera por medio de transiciones y estados.

Para construir el autómata este recibe una arreglo de tokens debidamente ordenado por el método del árbol para que se pueda generar el AFN correspondiente. Para la construcción del autómata se realiza con la ayuda de una pila

```
/**
 * CONSTRUCCIÓN DE AUTOMATA
 */
1 referencia
public void ConstruirAutomata(ArrayList arrayList)
{
    Stack pilaAFN = new Stack();
```

- Cerradura de Kleene Si viene la cerradura entonces este realiza un Pop del ultimo elemento introducido en la pila y se guarda en un autómata este autómata es introducido a la pila.

```
case "*":
    Automata automataKleene = CerraduraKleene((Automata)pilaAFN.Pop());
    pilaAFN.Push(automataKleene);
    this.Afn = automataKleene;
    break;
```

- Concatenación se sacan los últimos dos elementos introducidos en la pila se realiza la asignación de estados correspondientes y se vuelve a ingresar a la pila.

```
case ".":
    Automata concatenacionParametro = (Automata)pilaAFN.Pop();
    Automata concatenacionParametro2 = (Automata)pilaAFN.Pop();
    Automata automataConcatenacion = Concatenacion(concatenacionParametro2, concat
    pilaAFN.Push(automataConcatenacion);
    this.Afn = automataConcatenacion;
    break;
```

- Alternación: se sacan los últimos dos elementos introducidos en la pila se realiza la asignación de estados correspondientes y se vuelve a ingresar a la pila.

```
case "|":
    Automata alternacionParametro = (Automata)pilaAFN.Pop();
    Automata union_param2 = (Automata)pilaAFN.Pop();
    Automata automataAlternacion = Alternacion(alternacionParametro, union_param2);
    pilaAFN.Push(automataAlternacion);
    this.Afn = automataAlternacion;
    break;
```

- Interrogación: Se realiza el mismo procedimiento que la alternación solo que esta vez nosotros le mandamos Épsilon como otro parámetro.

```
case "?":
    Automata s = SimpleAFN("ε");
    Automata union_q1 = (Automata)pilaAFN.Pop();
    Automata qtion = Alternacion(union_q1, s);
    pilaAFN.Push(qtion);
    this.Afn = qtion;
    break;
```

Subconjuntos

Para la conversión del Autómata Finito No Determinista en uno Determinista se deberán aplicaron las siguientes operaciones:

- CerraduraE(Estado estado): Es el conjunto de estados alcanzables desde el conjunto de estados “X” solamente con el uso de transiciones ε este recibe como parámetro un estado y verifica las transiciones que puede alcanzar con épsilon incluyendo el mismo estado.

- Mover(X,a): Es el conjunto de estados alcanzables desde el conjunto de estados “X” utilizando transiciones “a”. Este método recibe una lista de estados HasShet para que estos no se repitan, y recibe un símbolo para poder realizar las transiciones con los estados, o puede si solo se evalua un estado puede hacer uso del método donde recibe el estado y el símbolo.